WHAT IS CLAIMED IS:

1   1. A method of checking the integrity of one or more input vectors to a digital

2 hardware block, comprising the steps of:

3   identifying a set of known bad input vectors for the digital hardware block;

4   and

5   training checking circuitry to selectively classify future input vectors to the

6   digital hardware block as either good or not good, using the set of

7   known bad input vectors.

8   2. The method of Claim 1 further comprising the step of classifying a new

9 input vector of the digital hardware block as not good, using the checking circuitry.

.1   3. The method of Claim 1 wherein said training step trains the checking

2 circuitry to classify as not good both future input vectors which are definitely faulty

3 and future inputs vectors which are potentially faulty.

1   4. The method of Claim 1 wherein said training step trains the checking

2 circuitry by feeding the set of known bad input vectors to a feedforward linear

3 associative memory neural network.

1   5. The method of Claim 4 wherein said training step includes the step of

2 creating a weight matrix using a discrete Hopfield network algorithm.

1   6. The method of Claim 5 wherein said creating step includes the step of

2 calculating the weight matrix W according to the equation

3
4
5

$$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

6 where $a_i^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input

7 vectors in the set of known bad input vectors, i is a row locator representing a

8 particular bad vector, and j is a column locator representing a bit location.

1   7. The method of Claim 6 wherein said classifying step includes the step of

2   calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new input

3   vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$.

1   8. The method of Claim 7 wherein said classifying step further includes the

2   step of adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i

3   according to the equation

4
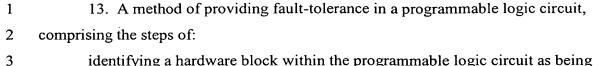$$\theta i = -\tfrac{1}{2} \sum_{j=1}^{K} w_{ij}$$

7   where K is the total number of bits in a vector.

1   9. The method of Claim 8 wherein said classifying step further includes the

2   step of processing each of the adjusted elements by a respective one of a plurality of

3   non-linear units such that, when a given adjusted element is positive, an output of the

4   corresponding non-linear unit is 1 and, when a given adjusted element is not positive,

5   the output of the corresponding non-linear unit is 0.

1   10. The method of Claim 2, further comprising the step of executing a

2   software work-around for the new input vector in response to said classifying step.

1   11. The method of Claim 1, further comprising the step of updating the

2   checking circuitry online.

1   12. The method of Claim 11 wherein:

2   the checking circuitry includes a feedforward linear associative memory

3   neural network having a weight matrix W; and

4   said updating step includes the step of reconfiguring the weight matrix W

5   using one or more additional bad input vectors.

1 13. A method of providing fault-tolerance in a programmable logic circuit,

2 comprising the steps of:

3   identifying a hardware block within the programmable logic circuit as being

4   faulty;

5   providing a software work-around for the faulty hardware block; and

6   training checking circuitry to selectively classify future input vectors to the

7   faulty hardware block as either good or not good, using a set of known

8   bad input vectors.

1 14. The method of Claim 13 further comprising the steps of:

2 handling a new input vector using the faulty hardware block;

3 classifying the new input vector as not good, using the checking circuitry;

4 executing a software work-around for the new input vector; and

5 in response to said classifying step, blocking an output of the faulty hardware

6   block corresponding to the new input vector, and accepting an output

7   of the software work-around corresponding to the new input vector.

1 15. The method of Claim 13 wherein said training step trains the checking

2 circuitry to classify as not good both future input vectors which are definitely faulty

3 and future inputs vectors which are potentially faulty.

1 16. The method of Claim 13 wherein said training step trains the checking

2 circuitry by feeding the set of known bad input vectors to a feedforward linear

3 associative memory neural network having a weight matrix W calculated according to

4 the equation

5

6 $$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

7

8 where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input

9 vectors in the set of known bad input vectors, i is a row locator representing a

10 particular bad vector, and j is a column locator representing a bit location.

1   17. The method of Claim 16 further comprising the step of updating the

2   checking circuitry online, by reconfiguring the weight matrix W using one or more

3   additional bad input vectors.

1   18. The method of Claim 16 wherein said classifying step includes the further

2   steps of:

3   calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the

4   new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$;

5   adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i

6   according to the equation

7   K

8    $\theta i = - \frac{1}{2} \Sigma\ w_{ij}$

9   j=1

10

11   where K is the total number of bits in a vector; and

12   processing each of the adjusted elements by a respective one of a plurality of

13   non-linear unit such that, when a given adjusted element is positive, an

14   output of the corresponding non-linear unit is 1 and, when a given

15   adjusted element is not positive, the output of the corresponding non-

16   linear unit is 0.

1   19. The method of Claim 18 further comprising the step of determining that a

2   vector constructed of the outputs of the non-linear units matches an entry in a memory

3   array storing the set of known bad vectors.

1   20. A circuit for checking the integrity of one or more input vectors to a

2   digital hardware block, comprising:

3   a memory array containing a weight matrix having elements which are based

4   on a set of known bad input vectors for the digital hardware block; and

5   means for selectively classifying future input vectors to the digital hardware

6   block as either good or not good, using the weight matrix.

- 18 -

1   21. The circuit of Claim 20 further comprising means for creating the weight

2   matrix using a feedforward linear associative memory neural network.

1   22. The circuit of Claim 20 wherein said classifying means classifies as not

2   good both future input vectors which are definitely faulty and future inputs vectors

3   which are potentially faulty.

1   23. The circuit of Claim 20 wherein the weight matrix W is created using a

2   discrete Hopfield network algorithm according to the equation

3

4    $$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

5

6   where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input

7   vectors in the set of known bad input vectors, i is a row locator representing a

8   particular bad vector, and j is a column locator representing a bit location.

1   24. The circuit of Claim 23 wherein said classifying means includes means

2   for calculating an output vector $a^{(m)}$ by multiplying the weight matrix W by the new

3   input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$.

1   25. The circuit of Claim 24 wherein said classifying means further includes

2   means for adjusting elements of the output vector $a^{(m)}$ by its respective thresholds θ_i

3   according to the equation

4    $$\theta i = -\frac{1}{2} \sum_{j=1}^{K} w_{ij}$$

5

6

7

8   where K is the total number of bits in a vector.

1   26. The circuit of Claim 25 wherein said classifying means includes a

2   plurality of non-linear units which respectively processes the adjusted elements such

3   that, when a given adjusted element is positive, an output of the corresponding non-

4   linear unit is 1 and, when a given adjusted element is not positive, the output of the
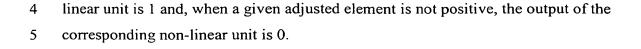
5   corresponding non-linear unit is 0.

1   27. The circuit of Claim 20 wherein said classifying means classifies a given

2   input vector as good or not good in less than 60 ns.

1   28. The circuit of Claim 20 further comprising mean for blocking an output of

2   the digital hardware block corresponding to a new input vector, and accepting an

3   output of a software work-around corresponding to the new input vector, in response

4   to said classifying means classifying the new input vector as not good.

1   29. The circuit of Claim 20 further comprising means for updating the weight

2   matrix online using one or more additional bad input vectors.

1   30. The circuit of Claim 27 further comprising:

2   an SRAM array;

3   a content-addressable memory for storing the known bad input vectors; and

4   address management means for using said SRAM to update both said content-

5   addressable memory and said weight matrix.

1   31. The circuit of Claim 20 wherein said classifying means classifies as not

2   good both future input vectors which are definitely faulty and future inputs vectors

3   which are potentially faulty.

1   32. The circuit of Claim 20 wherein said classifying means includes a content-

2   addressable memory for storing the known bad input vectors.

1   33. A programmable logic unit comprising:

2   a plurality of interconnected hardware blocks, at least one of said hardware

3   blocks being a faulty hardware block;

4   a feedforward linear associative memory (LAM) neural network checking

5   circuit which classifies input vectors to said faulty hardware block as

6   either good or not good;

7   a software work-around input; and

- 20 -

8   a selection circuit connected to said feedforward LAM neural network

9   checking circuit, for (i) blocking an output vector of the faulty

10   hardware block corresponding to a new input vector, (ii) enabling a

11   software work-around for the new input vector, and (iii) accepting an

12   output vector from said software work-around input corresponding to

13   the new input vector, as an output vector of the programmable logic
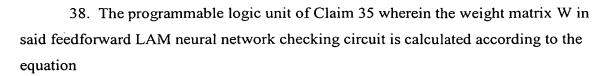
14   circuit.

1   34. The programmable logic unit of Claim 33 wherein said selection circuit

2   includes:

3   a software work-around enable signal output of said feedforward LAM neural

4   network checking circuit; and

5   a multiplexer having two inputs respectively receiving the output vector from

6   said faulty hardware block and the output vector from said software

7   work-around, and a select line which is connected to said software

8   work-around enable signal output of said feedforward LAM neural

9   network checking circuit.

1   35. The programmable logic unit of Claim 33 wherein said feedforward LAM

2   neural network checking circuit has a weight matrix whose elements are based on a

3   set of known bad input vectors for said faulty hardware block.

1   36. The programmable logic unit of Claim 35 wherein said feedforward LAM

2   neural network checking circuit updates the weight matrix online using one or more

3   additional bad input vectors.

1   37. The programmable logic unit of Claim 36 wherein said feedforward LAM

2   neural network checking circuit includes:

3   an SRAM array;

4   a content-addressable memory for storing the known bad input vectors; and

5   address management means for using said SRAM to update both said content-

6   addressable memory and said weight matrix.

1   38. The programmable logic unit of Claim 35 wherein the weight matrix W in

2 said feedforward LAM neural network checking circuit is calculated according to the

3 equation

4

5 $$w_{ij} = \sum_{m=1}^{M} (2a_i^{(m)} - 1)(2b_j^{(m)} - 1)$$

6

7 where $a^{(m)}$ is the set of known bad vectors, $a_i = b_j$, M is the number of bad input

8 vectors in the set of known bad input vectors, i is a row locator representing a

9 particular bad vector, and j is a column locator representing a bit location.

1   39. The programmable logic unit of Claim 38 wherein said feedforward LAM

2 neural network checking circuit includes:

3   means for calculating an output vector $a^{(m)}$ by multiplying the weight matrix

4   W by the new input vector $b^{(m)}$, that is, $a^{(m)} = Wb^{(m)}$;

5   means for adjusting elements of the output vector $a^{(m)}$ by its respective

6   thresholds θ_i according to the equation

7

8 $$\theta_i = -\tfrac{1}{2} \sum_{j=1}^{K} w_{ij}$$

9

10

11 where K is the total number of bits in a vector; and

12 a plurality of non-linear units which process respective elements of the weight

13   matrix to provide an output of 1 when a given adjusted element is

14   positive, and provide an output of 0 when a given adjusted element is

15   not positive.

1   40. The programmable logic unit of Claim 39 wherein said feedforward LAM

2 neural network checking circuit matches a vector constructed of the outputs of said

3 non-linear units with an entry in a content-addressable memory storing the set of

4 known bad vectors.

1   41. The programmable logic unit of Claim 33 wherein said feedforward LAM
2   neural network checking circuit classifies input vectors to said faulty hardware block
3   as either good or not good prior to said faulty hardware generating its output vector
4   corresponding to the new input vector.

1   42. The programmable logic unit of Claim 33 wherein said feedforward LAM
2   neural network checking circuit is physically located in a re-configurable hardware
3   redundancy block.